Using I&M personal geodatabases: Four examples

2/11/04 Eric Compas, revised 2/13/04 Sarah McGuire Bogen

With all inventory and monitoring data stored within a personal geodatabase, we have all spatial and non-spatial data stored in one centralized container. This container, however, doesn't immediately make the data usable or familiar to those of us used to ArcView, shapefiles, and DBF tables. This document describes how to do several familiar and common tasks within ArcGIS plus a few novel ones. While many of these tasks are fairly specific, they are meant to be examples of procedures that have more general application. For all examples, the Herbert Hoover National Historic Site exotic species I&M geodatabase is used.

Relational databases in ArcGIS

First, we need to look at the way that ArcGIS handles (or not) relational databases. For the tasks that we'll complete below, we need to know how to join tables with ArcGIS and how to connect to Access queries.

Joins and relates

ArcMap has two ways of connecting tables: joins and relates. Both use a link based on a common field between the two tables.

- Joins are one-to-one or many-to-one relationships where the fields of one table "show up" in the display of another table. For example, we could create a join on the **tblExotics** table to the **tluSpecies** table using **TSN** as the common field. The species' scientific and common name fields would then "appear" in the **tblExotics** table within ArcGIS.
- Relates are used to establish one-to-many relationships between tables. This means that a record in the source table may link to several records in the target table. Because there's more than one record, ArcMap does not show the fields of the related table with that of the sources as in a join. It only makes these related records available through selections. We won't be using relates within these examples.

To establish a join, right-click on the feature class or table (the source table) that you want the new fields to show up in. Select the menu options **Joins and Relates** and then **Join…**. The **Join Data** window will appear. Here you specify (**1.**) the join or common field within the source table, (**2.**) the join to table (the target table), and (**3.**) the join or common field within the target table. After clicking OK, the fields within the target field will be visible and available for additional joins.

Queries

ArcGIS does not natively see Access queries, and there are two less-than-ideal routes for seeing query results. We either have to go through an OLE DB Connection or by using a make-table query within Access to generate a temporary (but static) table. The advantage

of the OLE DB approach is that the queries are run dynamically and change as the underlying table data does. The make-table queries, however, only change when they're manually run. The distinct possibility exists for out-dated data to remain within these temporary make-tables. Therefore, the OLE DB Connection is preferred.

To make an OLE DB Connection, in ArcMap click on the **Add data** button and, within the **Look in:** box, select Database Connections. Below, we'll see an option to **Add OLE DB Connection**. Double-click this option to open the connection window that allows us to set the connection properties. Under the **Provider** tab, select the "Microsoft Jet 4.0 OLE DB Provider" and click on the **Next** button. Under the **Connection** tab, click on the button and select the personal geodatabase (the Access file). No other settings need to be changed. Click **OK** to establish the new connection. We'll now have a new OLE database connection, e.g. OLE DB Connection (2).odc, that we can open and view the Access tables and queries from ArcMap. These queries can then be used as part of a join.

Note: Relate functionality for OLE DB Connections is limited to the identity tool (no use of selected sets). See ESRI Technical Article #16466.

1. In ArcMap, how do I look at data within the geodatabase?

Adding the spatial and table data to an ArcMap session is only slightly different from adding coverages, shapefiles, or INFO tables. First, click the Add Data and move to the folder containing the geodatabase , e.g. HEHOExoPltGDB.mdb. Double-clicking the geodatabase will open it like a folder. Within, we'll see a combination of tables , feature datasets , and feature classes . Note that feature datasets contain additional feature classes, and we'll need to double-click the feature dataset to see them. Any of these data types can be selected and added to ArcMap. If a feature dataset is added, all of the feature classes with in the dataset will be added. Adding tables will allow them to be part of joins or relates. We need to switch to the **Source** tab of the table of contents in ArcMap to see the tables that we've added. Check the boxes next to any feature classes to show these spatial layers in the current map.

All of the examples below assume that we have added all of the feature classes and tables to an ArcMap session.

2. In ArcMap, how can I show (label or symbol) species at each location?

To get from our feature class to the species name, we need to negotiate several relationships within the geodatabase. For example, in the HEHO geodatabase we need to go from **fclExoPltPoint** to **tblExotics** and then to **tluSpecies**. We can do this within Arcmap by establishing several joints in the correct order. First, we need to join the **fclExoPltPoint** feature class to the **tblExotics** table using the **LocationID** field. Once this join is established, the fields of **tblExotics** are available within the **fclExoPltPoint** table. Second, establish another join on the **fclExoPltPoint** table to the **tluSpecies** table

using the **TSN** field. We can now see the fields of **tluSpecies** and can use them for labeling or symbolizing.

To label species common names, right-click the **fclExoPltPoint** feature class and select **Properties…**. In the Properties window, select the **Labels** tab and under **Text String**, select the field that contains the common name. Right-clicking the feature class again and selecting Label Features will add the common names to the map display.

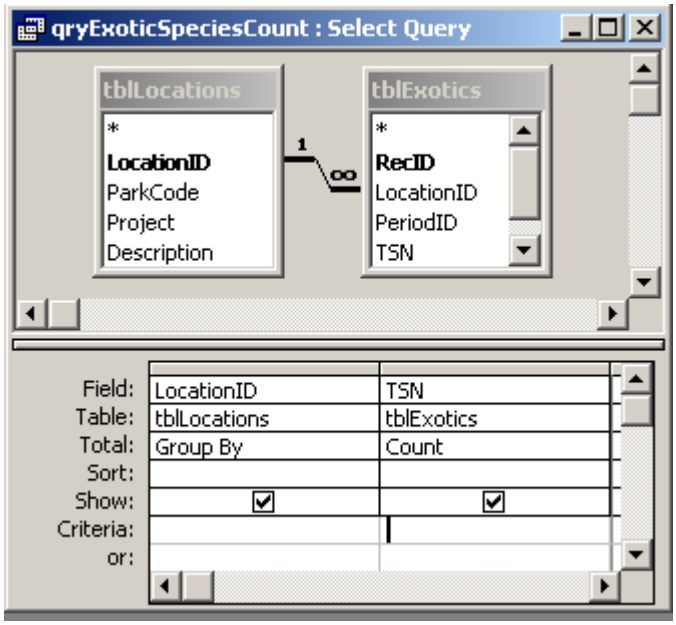
This procedure works well if there is only one species per data point because only the first species is displayed if there is more than one. If many or most of the locations have more than one species, we will need to use a query approach as in Question 3.

Note: The order in establishing these joins is important and joins can't be established to tables that are already part of another join.

3. *In ArcMap, how do I show the number of species at each location?*

To accomplish this task, we'll need to do more than just establish a join. We need to generate a query within Access that counts the number of species per location. We can then join this query to a feature class to display the results.

In Access, create a new select query in design view. For HEHO, we'll need to add the tables that contain information about locations (tblLocations) and exotic species records for each (tblExotics) as seen below:



First, add the LocationID field to summarize on and add the TSN field to count. To calculate a count per location:

- Click on the **Totals** button 

- Set LocationID to “Group By”
- Set TSN to “Count”. We can then
- Test run the query to make sure the results are as expected.
- Save the query with a descriptive name such as “CountbyTSN” before closing it.

Back in ArcMap, add the feature class for the locations considered in the query (can use points, lines, or polygons) and add the new query.

- Click on the **Add Data** button and in the **Look in:** box, choose **OLE DB Connection.odc**.
- Choose the feature class considered in the query and the new query.

Using a join on the feature class to our query with LocationID as a common field, we now have access to the CountOfTSN field to display or analyze the number of species per location.

We can use the same procedure to show the results of other and more complicated queries.

4. How do I summarize data for features based on some other polygon layer, e.g. management units or vegetation?

Here, we’d like to calculate some statistic or summary for our data based on where points, lines, or polygons fall in some other layer. For example, we might want to know the number and type of exotic species for each management unit in a park.

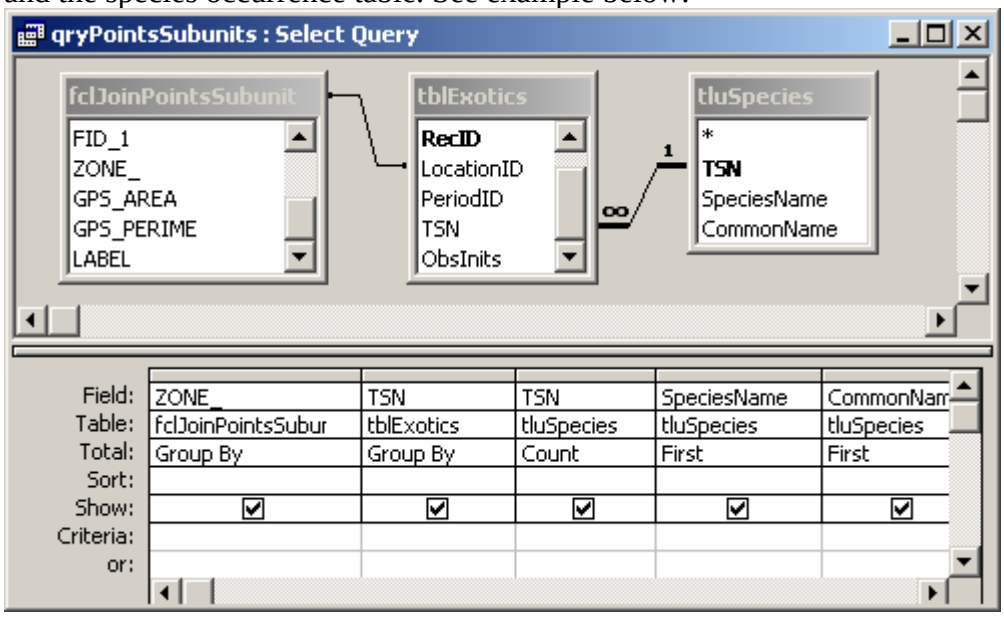
To accomplish this task, we’ll use both ArcGIS and Access: ArcGIS to spatially join our layers and Access to summarize the results. A spatial join relates two tables based on common locations instead of common fields. For example, we could spatially join a point layer to a polygon layer based on the points that fall within each polygon. ArcGIS gives us several options for spatial joins depending on the feature types.

To create a spatial join, add both the source data locations—in this example the **fdsExoPltPoints** feature data set—and a polygon summary layer, such as the park boundary, to ArcMap. Right-click the **fdsExoPltPoints** feature data set and select **Joins and Relates** and then **Joins…**.

Within the **Join Data** window where it asks what we’d like to join, select the option “Join data from another layer based on spatial location.” Under layer to join (**1.**), select the polygon summary layer and for option **2.**, choose the option for the point to “fall inside” the polygon. Note that the options for **2**. may change depending on our summary and on the type of features involved in the join. For option **3.**, choose the output layer (shapefile or feature class) to be created. To more easily use the results within Access, create a feature class within the same geodatabase as our point file. Click **OK** to execute the join. Close the ArcMap session in order for us to get full access to the database.

Open the geodatabase within Access. We're using Access instead of ArcMap's summarize capability since the relationship is (most of the time) a one-to-many. Each location within our point feature class may (and does in many cases) have more than one species for each point. If we did this analysis in ArcMap, we'd only get the first species.

Within Access, we need to create a new select query to summarize our data for each polygon unit. Include in the query the table from the just created spatial join (**fclJoinPointsSubunit**) and the tables that contain species occurrence (**tblExotics**) and species names (**tluSpecies**). We'll need to establish the join between the spatial join table and the species occurrence table. See example below:



To summarize, we'll need to add total via the **Totals** button Σ and set the polygon unit (here ZONE_) and one of our species fields (here TSN) as our "Group By" summarizing fields. (For those of you used to doing "frequency" tables in Arc/Info or ArcView, this is basically the same process.) We've included the TSN field a second time to count the number of occurrences of each species for our unit-species summary. The "First" occurrence of SpeciesName and CommonName allow us to interpret the results of the query. Saving and running the query will result in a table that's our summary or frequency table for each unique zone-species combination.

To produce a final output, we can create an Access report to group the results by unit and produce some summary statistics, e.g. total for each zone.

Using criteria within this query, we could modify these results to examine only one species and produce the same summary.

Note: All of these steps could also be carried out within ArcMap, but the results would only be valid if there were only one species per points. Using Access allows us to account for this eventuality.

5. Similar to Example #4, how can I summarize the areas of polygons that overlap another polygon layer?

Like example #4, we'd like to summarize one polygon layer based on another polygon layer, but we need to summarize on some area calculation. For example, we may need to know the proportion or total area of an exotic species distribution on a vegetation class or management area polygon layer. Here, the spatial join created above isn't sufficient; we need to "intersect" these polygon layers to get at these area values.

In an ArcMap session with both polygon layers added, we first want to add an area field (if it doesn't already exist) to the polygon layer that we'll be calculating a proportion of. For our example, that'll be the **hehopmup** (management units) layer. Open the layer's attribute table, add a field called "ZoneArea", and calculate a value for the field using the following:

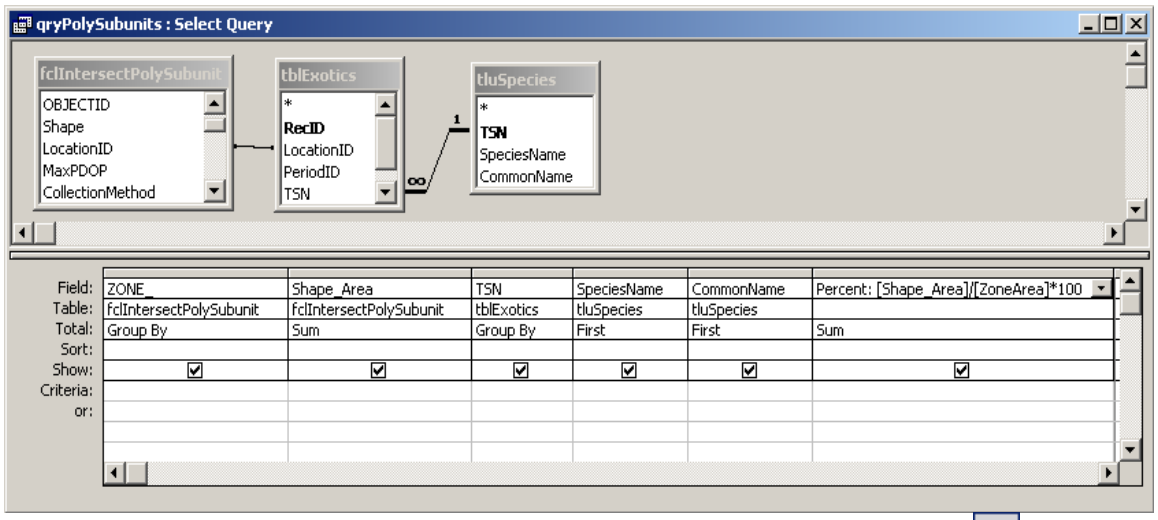
1. Select "shape" as field to calculate from.
2. In calculate dialog, check "Advanced."
3. Add the following code:

```
Dim Output as Double
Dim pArea as IArea
Set pArea = [shape]
Output = pArea.area
```

4. Add "Output" to "ZoneArea =" box.
5. Click "OK" to calculate the area values.

Next, we intersect our two polygon layers by selecting **Tools** -> **GeoProcessing wizard…**. Within the wizard that opens, select the option to "Intersect two layers." In the next screen, select the two polygon layers that we'd like to intersect—the order doesn't matter. For our example, these will be **HEHO** and **hehopmup**. Pay attention to the "Use selection features only" option—we may want this option selected if we're looking a specific species or class. For this example, we will not be working with selected sets. Also, specify where the resulting layer will be saved. We'll need to create a feature class in our personal geodatabase to use the resulting figures within Access. We'll call that **fclIntersectPolySubunit**. Clicking "Finish" will create the new feature class.

Now, open the geodatabase within Access and create a new query in design view. We'll be creating something like the following:



Here, we've created a select query with totals (remember the **Totals** button Σ) and added three tables: **fclIntersectPolySubunit**, **tblExotics**, and **tluSpecies**. We'll need to add the relationship between **fclIntersectPolySubunitb** and **tblExotics** by dragging the LocationID field from one to the other. To this query, we've added the "Zone_" field and "TSN" field to group our query results by. This means will be calculating totals for unique combinations for each zone and exotic species. To show the area total, we need to add the "Shape_Area" field and set it "Total:" option to "Sum." We can also add the SpeciesName or CommonName to our query. Setting the "Total:" option to "First" will grab the first occurrence of these fields. Lastly, to calculate the proportion of the management zone covered by each species, we'll add a calculated field to our query. Here, we've called it "Percent" and divided the Shape_Area by the ZoneArea field and multiplied by 100 to get a percent. Since this calculation is for each occurrence of a species within a zone, we need to set the "Total:" option to "Sum." This we give us the total percent. Running the query will yield a table with area totals and proportions of exotic species coverages within each management area. Like example #4, we could make these results more presentable by creating a report from the results.

Note: since the polygon extents within the exotic species polygon layer overlap. The sum of these percentages does not equal the total percent of the management area covered by exotic species.

Example relationships

Here is a diagram of the relationships with the Herbert Hoover exotics plant personal geodatabase that may be helpful in understanding the joins established above: